Hidden Markov Model & It's Application.

Presentation Outline

Hidden Markov Model

- The Hidden Markov Model is dual Stochastic Process, where one of the underlying process is Hidden.

- The hidden process is a Markov chain moving from one state to another but cannot be observed directly.

- The other process is observable but its movement depends upon hidden state.

- Hidden Markov model is a branch of machine learning. it’s useful in solving problem related to sequence.

HMM Example

●○○ A Short History

Markov chain is discrete stochastic process, where probability of event occurring only depend upon immediate previous event.

●●○ Popular Use Case

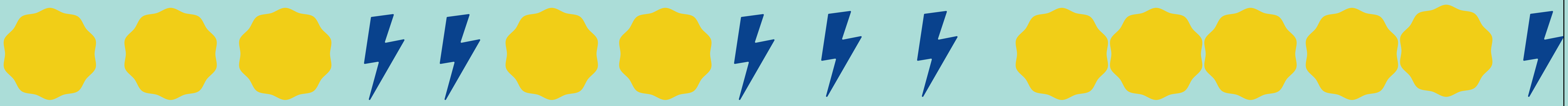
Primarily it's use where there are sequence of event which take place. One of the popular use case is in weather forecasting.

●●● Two States Example

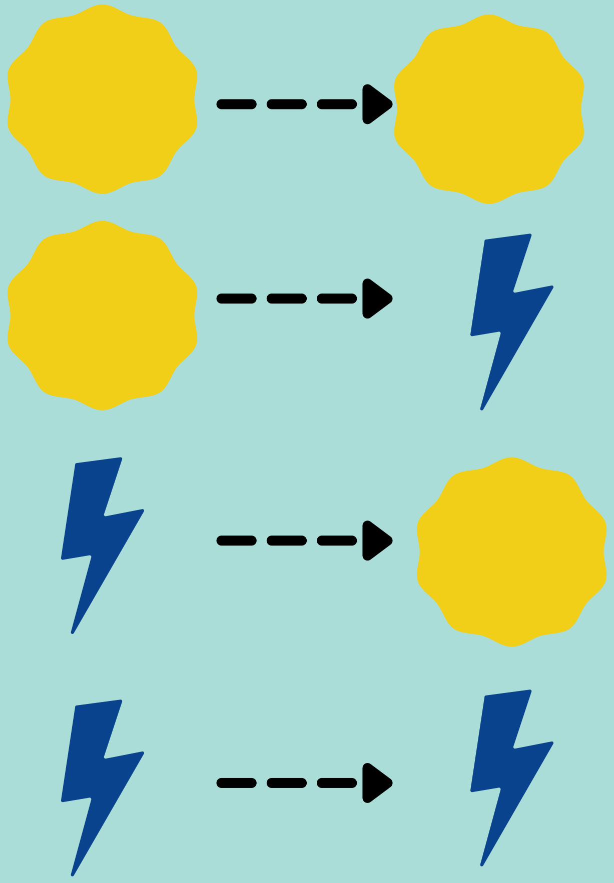
Let's analyse two states example of weather, here two states means it can be sunny or rainy.

HMM Example : Weather Data

Sunny or Rainy Observations of 15 days.



Transition Probabilities

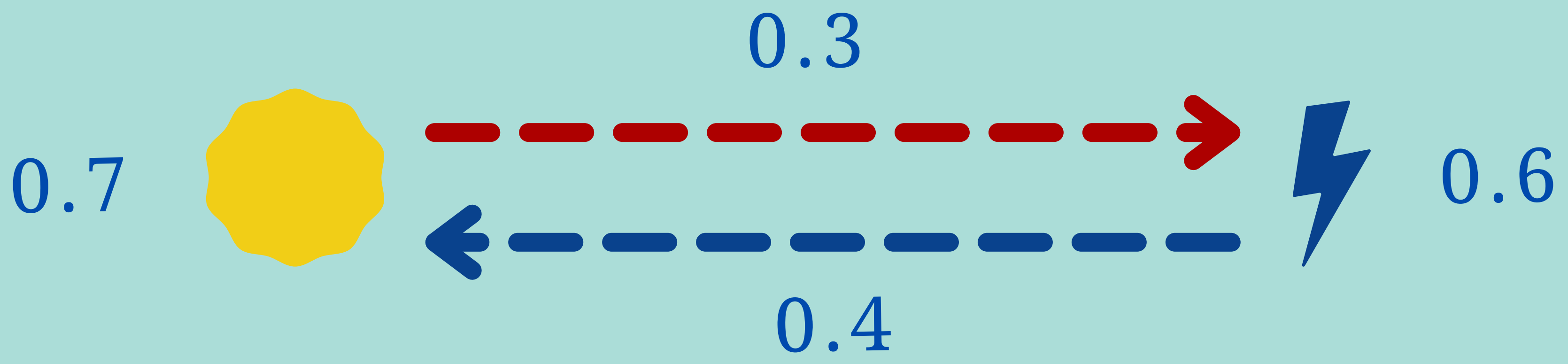


7/10 = 0.7

3/10 = 0.3

2/5 = 0.4

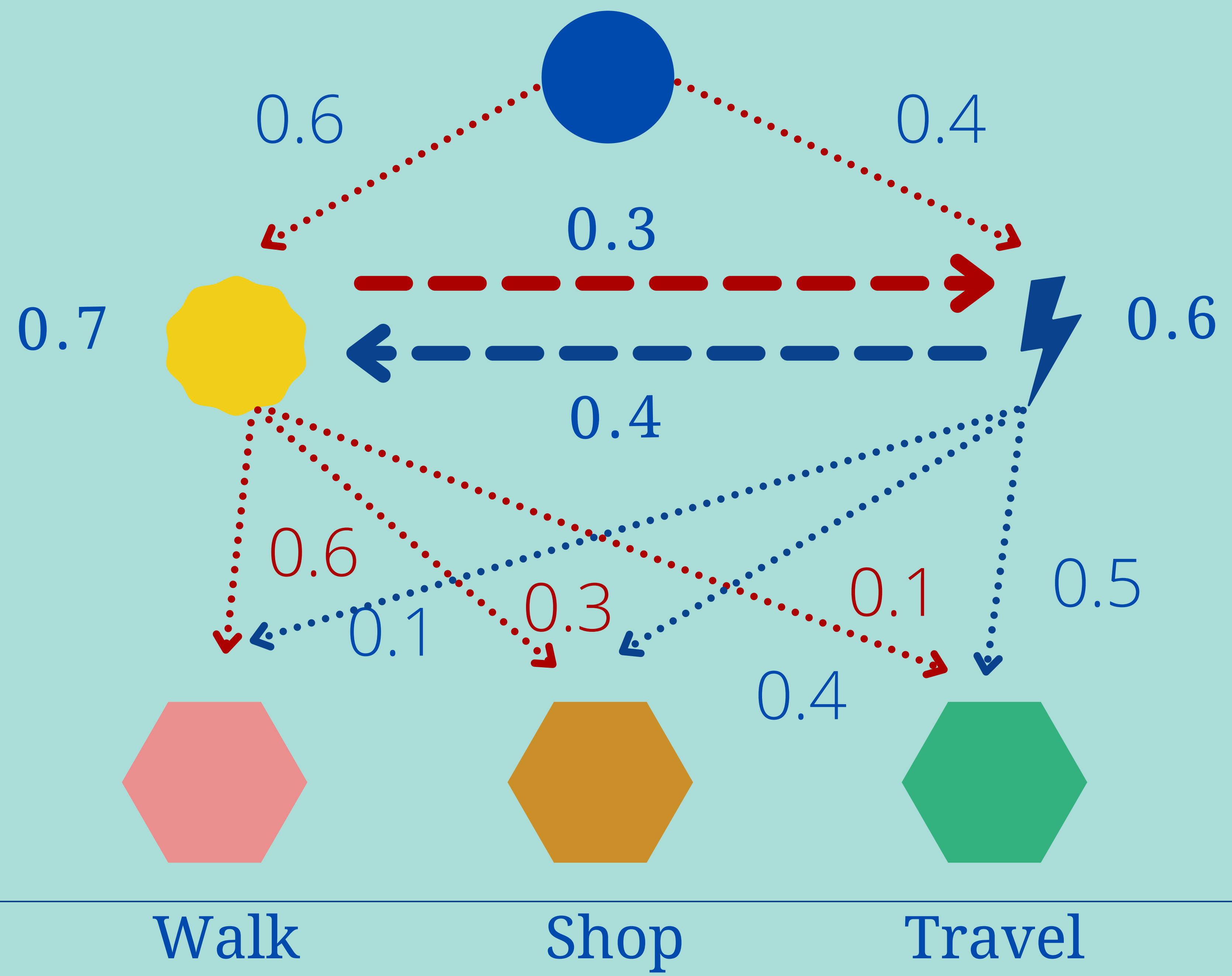
3/5 = 0.6



HMM Example : Weather Data

The Application of HMM would be based on real action, let's say there are three actions person can do depend upon weather like "walk", "Shop", "Travel"

Initial Probability : Pi



HMM Example : Terminology & Calculations

- X0 = Initial Probability Distribution :Probability that chain will start at some state
- X = Hidden State (Rain / Cloud)
- Y = Observables (Walk, Shop, Travel)
- a = Transition Probability (Moving from Rain to Sunny and vice versa).
- b = Emmission Probability (Observation being generated from State)

Person goes for walk, first day it was rainy and second day it was sunny.

P((walk,walk),(rainy, sunny)) :

P((walk|rainy) * P(walk|sunny) * P(sunny|rainy) * P(rainy)

0.1 * 0.6 * 0.4 * 0.4 = 0.0096

HMM: Three Elements

- In order to model HMM we need to run through three algorithms.
- **Forward-Backward Algorithm:** It's evaluation phase where computation of probability of Observation Sequence.
- **Baum-Welch Algorithm**: It's learning phase for determination of parameter of model.
- **Viterbi Algorithm:** Decoding the most probable state sequence.

HMM: Three Elements : Forward-Backward Algorithm

- The Forward-Backward algorithm computes the posterior (updated probability of an event occurring after taking into consideration new information.) marginals of all hidden state variables.

- The Algorithm uses two passes, the first one goes forward in time and the second pass goes backward.

- The First pass computes set of probabilities which provides probability of ending up in particular state.

- The Second pass computes a set of backward probabilities which provide the probability of observing remaining observation given any starting point.

- Basically this algorithm is used to find the most likely state for any point in time.

HMM: Three Elements : Baum-Welch Algorithm

- This algorithm deals with unknown parameters of a hidden markov model.
- It's case of E-M Algorithm (Expectation Maximization) which is a method to find maximum estimation.
- The "E" part of Expectation re-estimate the pi given current HMM parameter.
- The "M" part of Maximization, re-estimating HMM parameter given current pi state.
- In Conclusion, the baum-welch algorithm attempt to find model that assigns the training data the highest likelihood.

HMM: Three Elements : Viterbi Algorithm

- The most useful algorithm when one wants to calculate the most likely path through the state transition.
- The observation made by this algorithm is that at any state at time T, there is only one most likely path to that state.
- Using this algorithm we can find the most likely sequence of hidden states given the sequence of observations.

HMM Application in Python

- So far we have observed how this entire machine learning algorithm works.
- The Example which we have gone through (Weather Forecasting) is of discrete in nature.
- Stock returns are continuous in nature and random noise is already present there.
- In order to model random noise we need to use Gaussian model and it takes two input mean and variance.
- Observed set of data is of NIFTY (National Stock Exchange of India) Index.
- Data set is for last 10 years (2011-2020) data of 15 minutes interval.

HMM Application in Python

- Code here is very simple unoptimized code for illustration perspective.

```
# Import necessary libraries to perform HMM
import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import hmmlearn as hmm
from hmmlearn.hmm import GaussianHMM
```

```
raw_data.head()
```

	Ticker	Date/Time	Open	High	LOW	Close	Volume
0	NIFTY-I	03-01-2011 09:15	6202.10	6207.35	6195.00	6198.25	1347650
1	NIFTY-I	03-01-2011 09:30	6198.25	6204.90	6193.65	6200.00	618550
2	NIFTY-I	03-01-2011 09:45	6200.05	6202.70	6196.15	6199.50	260950
3	NIFTY-I	03-01-2011 10:00	6199.50	6201.90	6195.50	6195.60	172050
4	NIFTY-I	03-01-2011 10:15	6195.80	6198.00	6186.00	6187.00	459500

HMM Application in Python

```
# Extract required details for modelling.
dates = np.array(raw_data['Date/Time'])
close_price = np.array(raw_data['Close'])
volume = np.array(raw_data['Volume'])
```

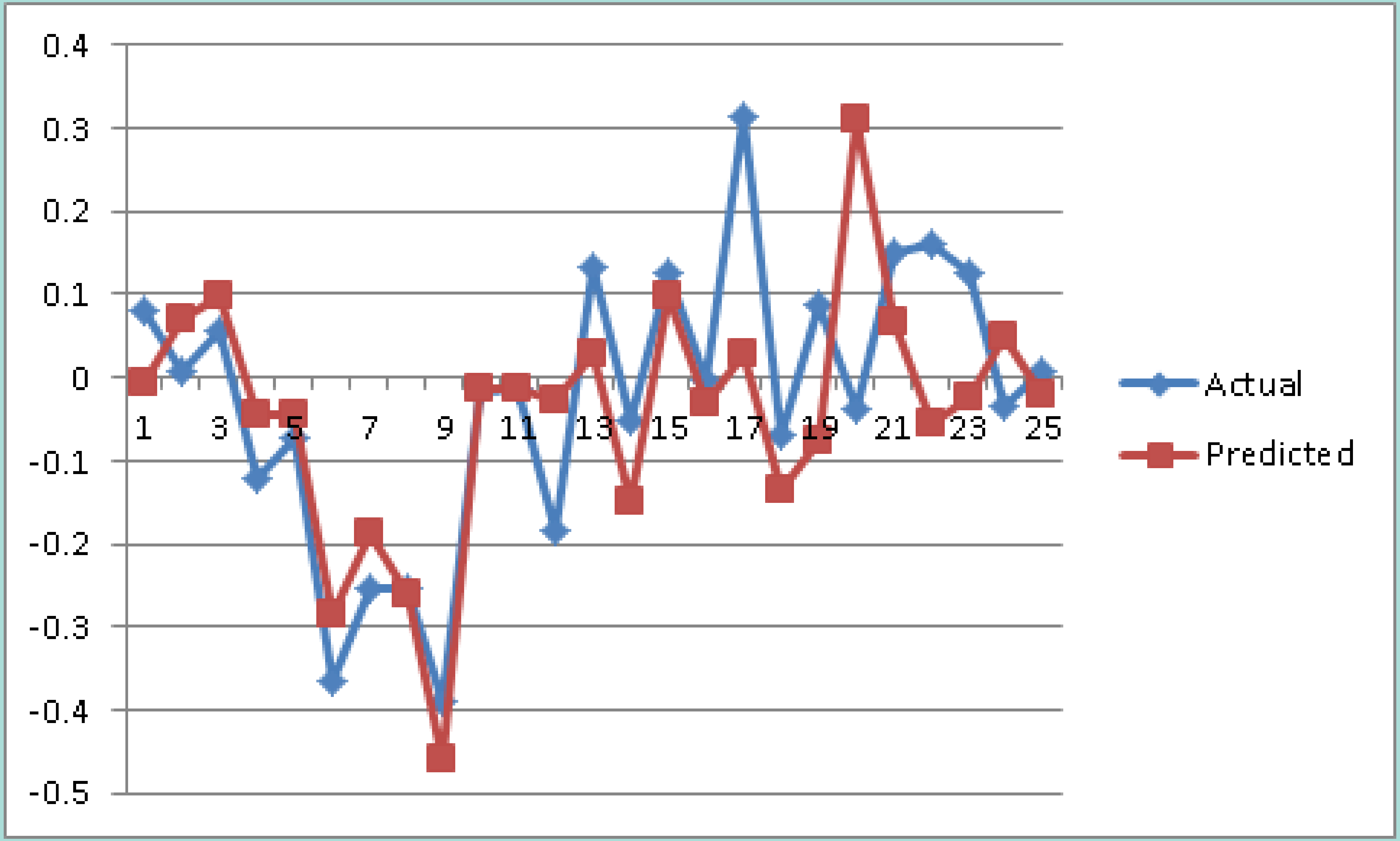
```
# Take difference of closing price and compute rate of change
diff_percetage = 100.0*np.diff(close_price)/close_price[:-1]
dates =dates[1:]
volume = volume[1:]
```

```
# Stack percetage difference and volume for columnwise for training.
X = np.column_stack([diff_percetage,volume])
```

```
# create and train Gaussian HMM
print("\nTraining HMM...")
model = GaussianHMM(n_components=5, covariance_type="diag", n_iter=1000)
model.fit(X)
```

```
# Generate data using model
num_samples = 25
samples,_=model.sample(num_samples)
plt.plot(np.arange(num_samples),samples[:,0], c='blue')
plt.show()
```

HMM Final Prediction and Actual Values.



Acknowledgment:

Various paper and webpages helped to complete this project.

- https://medium.com/analytics-vidhya/hidden-markov-model-a-statespace-probabilistic-forecasting-approach-in-quantitative-finance-df308e259856
- https://towardsdatascience.com/probability-learning-vi-hidden-markov-models-fab5c1f0a31d
- Experimental Mathematics: hidden Markov models.pdf
- https://rubikscode.net/2018/10/29/stock-price-prediction-using-hidden-markov-model/